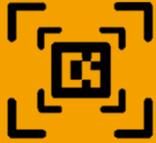
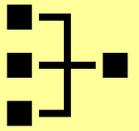


FALCON

Vision-Based Dynamic Landing System for UAVs on Moving Platforms



Real-time localization and trajectory prediction for automated drone landings



Control and filter algorithms for swift yet precise maneuvers regardless of adverse conditions



Problem Statement

Most drones today can only land on static surfaces. However, real-world applications like maritime search-and-rescue, autonomous deliveries, or military deployments often require landing on moving platforms (e.g., a boat, ground vehicle, or even another drone). This poses challenges like uncertain platform trajectory, variable environmental conditions, and the need for continuous visual feedback.



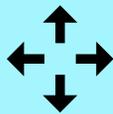
FALCON

Our vision-based drone landing system explores autonomous robotics to create **reusable and modular** software components with the intention of being **real-time and easily extensible**. FALCON presents users with a **python-based GUI** that abstracts the backend complex algorithms. Users can **adapt FALCON's capabilities into any drone, fine tune parameters and simulate in real time**.

Features and Capabilities



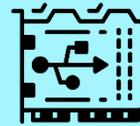
1. Real-time marker recognition & distortion handling



2. Motion approx. using marker frame sequences



3. Sensor fusion using Inertial Measurement Unit



4. PID-based Control logic and algorithm with noise filtering



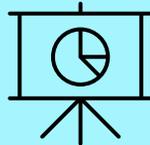
5. Marker loss recovery logic using estimated trajectory



6. Front End GUI for Users to interface with FALCON



7. Configurable PID control parameters through GUI



8. Configurable live feed from Camera with Overlay for visualization



9. Fail-safe landing mode and warning system



10. Software test harness using simulated mock data

Software Engineering Practices

Version Control: All code tracked on GitHub using Issues, Pull Requests, Branches, milestone labels, tags, assignees.

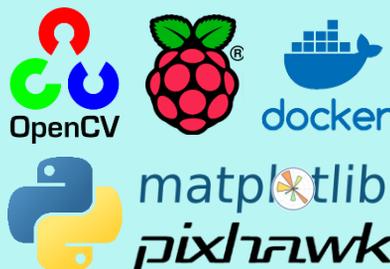
Documentation: Inline code comments, README docs, architecture diagrams.

Modular Design: Separation of Concerns (SoC) across sensors, vision, control, and UI.

Testing: Unit tests and integration tests using mocked sensor and tag data.

CI/CD: GitHub Actions for continuous integration with basic testing pipeline

Tech Stack



Project By:

Joel Ku and Kenneth Wong Cun Wi

FALCON for Users:

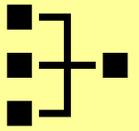
[HTTPS://GITHUB.COM/KUROKISHIS92/FALCON.GIT](https://github.com/kurokishis92/falcon.git)

FALCON

Vision-Based Dynamic Landing System for UAVs on Moving Platforms



Real-time localization and trajectory prediction for automated drone landings



Control and filter algorithms for swift yet precise maneuvers regardless of adverse conditions



Problem Statement

Most drones today can only land on static surfaces. However, real-world applications like maritime search-and-rescue, autonomous deliveries, or military deployments often require landing on moving platforms (e.g., a boat, ground vehicle, or even another drone). This poses challenges like uncertain platform trajectory, variable environmental conditions, and the need for continuous visual feedback.



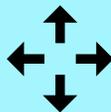
FALCON

Our vision-based drone landing system explores autonomous robotics to create **reusable and modular** software components with the intention of being **real-time and easily extensible**. FALCON presents users with a **python-based GUI** that abstracts the backend complex algorithms. Users can **adapt FALCON's capabilities into any drone, fine tune parameters and simulate in real time**.

Features and Capabilities



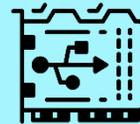
1. Real-time marker recognition & distortion handling



2. Motion approx. using marker frame sequences



3. Sensor fusion using Inertial Measurement Unit



4. PID-based Control logic and algorithm with noise filtering



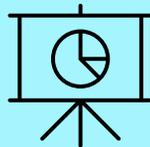
5. Marker loss recovery logic using estimated trajectory



6. Front End GUI for Users to interface with FALCON



7. Configurable PID control parameters through GUI



8. Logging/replay for debugging & visualization



9. Fail-safe landing mode and warning system



10. Software test harness using simulated mock data

Software Engineering Practices

Version Control: All code tracked on GitHub using Issues, Pull Requests, Branches, milestone labels, tags, assignees.

Documentation: Inline code comments, README docs, architecture diagrams.

Modular Design: Separation of Concerns (SoC) across sensors, vision, control, and UI.

Testing: Unit tests and integration tests using mocked sensor and tag data.

CI/CD: GitHub Actions for continuous integration with basic testing pipeline

Tech Stack



Project By:

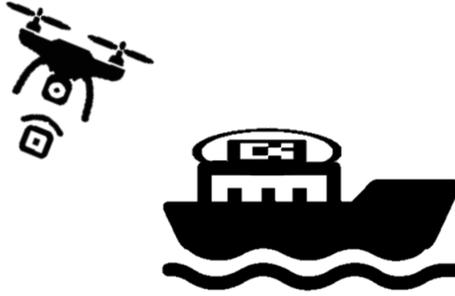
Joel Ku and Kenneth Wong Cun Wi

FALCON for Users:

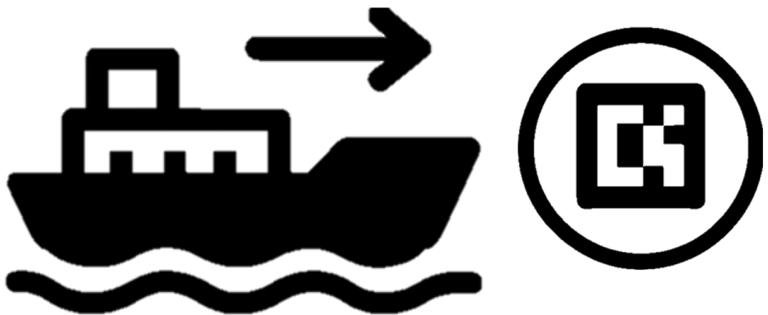
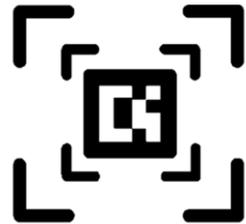
[HTTPS://GITHUB.COM/KUROKISHIS92/FALCON.GIT](https://github.com/kurokishis92/falcon.git)

FALCON

Vision-Based Dynamic Landing System
for UAVs on Moving Platforms



About



Problem Statement

Features

Tech Stack

User Interface

Evaluation